

We Claim:

Sub #1 1. In a computer, a method of enhancing scalability of server applications, comprising:

5 executing an application component under control of an operating service, the application component having a state and function code for performing work responsive to a call from a client; and

destroying the state in response to an indication from the application component that the work is complete without action by the client.

10 2. The method of claim 1 wherein the operating service retains a reference to the application component and the step of destroying the state comprises releasing the reference to the application component by the operating service.

15 3. The method of claim 1 wherein the step of destroying the state comprises resetting the state of the application component.

4. The method of claim 1 further comprising:  
performing the step of destroying the state upon a next return of the application component from the client's call following the indication from the application component that the work is complete.

20 Sub E 5. In a computer, a computer operating environment for scalable, component-based server applications, comprising:  
a run-time service for executing an application component in a process, the application component having a state and implementing a set of functions;

25 an instance creation service operative, responsive to a request of a client, to return a reference to the application component through the run-time service to the client, whereby the client calls functions of the application component indirectly through the run-time service using the reference to initiate work by the application component; and

30 the run-time service being operative, responsive to an indication from the application component that the application component has completed the work for the client, to destroy the application component's state on the

08959149 102897  
268201 64765680

application component returning from a call by the client without action by the client.

6. The computer operating environment of claim 5 wherein the indication is a call from the client to commit or abort a transaction encompassing the work.

7. The computer operating environment of claim 5 wherein the application component initiates the indication before returning from the call by the client, whereby the application component's state is destroyed immediately on return from the client's call without further action by the client.

8. The computer operating environment of claim 5 further comprising:

a component context associated by the run-time service with the application component and providing an interface having a member function that the application component calls to initiate the indication.

9. The computer operating environment of claim 8 wherein the application component performs the work within a transaction and wherein calling the member function of the component context causes the transaction to abort.

10. The computer operating environment of claim 8 wherein the application component performs the work within a transaction and wherein calling the member function of the component context permits the transaction to commit.

11. The computer operating environment of claim 5 wherein the run-time service holds a reference to an instance of the application component, and destroys the application component's state by releasing the reference to the instance.

12. The computer operating environment of claim 5 wherein the run-time service destroys the application component's state by resetting the state.

13. In a computer, a method of encapsulating state of processing work for a client by a server application in a component with improved scalability, comprising:

sub  
C37  
30

08959149 102897  
63207 6465680

encapsulating function code and a processing state for the work in a component;

providing a reference through an operating service for a client program to call the function code of the component to initiate processing of the work by the component;

receiving an indication from the component that the work by the component is complete; and

discarding the processing state of the component responsive to the component indicating completion of the work.

10 14. The method of claim 13 further comprising:

performing the step of discarding the processing state upon a next return of the component from a call of the client program following the indication from the component that the work is complete.

15 15. The method of claim 13 wherein the step of receiving the indication comprises:

providing a context object containing data representing a context of the component and having an integration interface for receiving a call of the component to indicate that the work by the component is complete.

20 16. The method of claim 15 wherein the call of the component to the integration interface of the context object further indicates that a transaction encompassing the work is to be committed.

17. The method of claim 15 wherein the call of the component to the integration interface of the context object further indicates that a transaction encompassing the work is to be aborted.

25 18. In a computer, a system service for providing an execution environment for scalable application components, comprising:

code responsive to a request from a client program to create an application component for returning to the client program a reference through the system service to the application component;

30 code responsive to a call from the client program using the reference for initiating processing of work by the application component, the application component producing a processing state during processing the work;

08959149 463207 6465680

code for receiving an indication from the application component that processing by the application component of the work is complete; and

code for destroying the processing state of the application program without action from the client program.

5           19.    The system service of claim 18 further comprising:

code for producing an instance of the application component and retaining a reference to the instance, the instance containing the processing state; and

10           wherein the code for destroying the processing state comprises code for releasing the reference to the instance without action from the client program to thereby cause the processing state to be destroyed.

20.    The system service of claim 18 wherein the code for destroying the processing state comprises:

15           code for resetting the processing state to an initial state of the application component.

7  
32  
22  
22  
22

08959149 102897  
268201" 64T65680